

VISUAL BASIC 6.0

SETYO BUDI, M.KOM

ARRAY DAN PROCEDURE

Array

- **Array** adalah kumpulan, merupakan variabel memori yang digunakan untuk menyimpan sekumpulan data dalam tipe yang sejenis.
- Data-data pada array disimpan dalam urutan tertentu yang dinamakan index
- Sehingga untuk merujuk pada lokasi tertentu atau element array tertentu maka kita menentukan nama array dan posisi elemen array yg diidentifikasi menggunakan index
- Setiap nomor index dalam array dialokasikan dalam ruang memori tertentu, sehingga pengguna harus menghindari pernyataan array yang ukurannya lebih besar daripada yang dibutuhkan.
- Sebagai contoh jika x adalah variabel array maka digambarkan sbb

Array

Variabel x

Data ke 1	Elemen/index ke 1
Data ke 2	Elemen/index ke 2
.....	
Datan ke	Elemen/index ke n

Array dapat didefinisikan dari tipe data dasar termasuk tipe primitif, maupun tipe user-defined, dan variabel objek

- Sedangkan elemen individual array adalah semua tipe data yang sama

Array

Terdapat dua tipe array pada visual basic

- **Fixed Size Array** : ukuran array selalu tetap dan tidak berubah selama program di eksekusi.
- **Dynamic Array** : ukuran array dapat berubah saat –
Dynamic Array : ukuran array dapat berubah saat program runtime selama program di eksekusi

Fixed Size Array

- Atau sering disebut **array statik** merupakan variabel array dimana **ukuran array tidak dapat berubah saat runtime.**
- Bentuk definisi variabel array ini sbb :

Dim bulan(11) as integer

Ketika kita mendefinisikan variabel array dengan menentukan sebuah nilai index maka index array akan dimulai dari 0 hingga maksimal nilai index.

- Jika kita menginginkan nilai array tidak diawali dengan 0 maka kita dapat menambahkan **option base** pada definisi array sbb:
- Sehingga definisi variabel array menjadi sbb :

Dim bulan(1 to 12) as integer

Private Sub Form_Load()

```
Dim bulan(1 To 12) As String
```

```
bulan(1) = "Januari"
```

```
bulan(2) = "Februari"
```

```
Debug.Print bulan(2)
```

End sub

Hasilnya : Februari

Dynamic Array

- Pada dasarnya kita dapat membuat definisi variabel array baik dalam bentuk array statik / fixed-size atau array yang dinamik.
- Ketika array didefinisikan statik maka kita perlu menyertakan nomor index dan harus diketahui pada saat kompilasi.
- Sehingga kompilator dapat menyisihkan jumlah array yg diperlukan pada pernyataan dim dan dengan argumen yang konstan
- Dengan array statik compiler tidak memperbolehkan perubahan ukuran array saat aplikasi di eksekusi.
- Untuk mengatasi permasalahan tersebut maka perlu mendefinisikan array dalam bentuk array dinamis
- Ketika kita mendefinisikan array dinamis maka kita tidak menyertakan ukuran array pada pernyataan dim

Dynamic Array

- Selanjutnya dapat dilakukan perubahan array menggunakan pernyataan **ReDim**

Contoh

Dim kota() As String

Definisi Dinamic Array tanpa menyertakan index array

ReDim kota(10) As String

Selanjutnya menggunakan pernyataan **ReDim** untuk menentukan ukuran array

Perhatikan Contoh Berikut

```
Dim kota() As String
```

```
ReDim kota(10) As String
```

```
kota(0) = "Semarang"
```

```
Debug.Print kota(0)
```

Hasil

Semarang

Perhatikan Contoh Berikut

```
Dim kota() As String
```

```
ReDim kota(10) As String
```

```
kota(0) = "Semarang"
```

```
ReDim kota(4) As String
```

```
Debug.Print kota(0)
```

Ketika Dinamic Array mengalami redefinisi maka nilai pada array akan hilang, agar nilai pada array kekal pada saat redefinisi maka pada redefinisi perlu menambahkan keyword Preserve

Perhatikan Contoh Berikut

```
Dim kota() As String
```

```
ReDim kota(10) As String
```

```
kota(0) = "Semarang"
```

Perhatikan Perhatikan Contoh Contoh Berikut Berikut

```
ReDim Preserve kota(4) As String
```

```
Debug.Print kota(0)
```

Dengan keyword Preserve nilai pada array tidak hilang saat redefinisi

Hasilnya : Semarang

Array Multidimensi

- Array dapat memiliki beberapa dimensi
- Penggunaan umum array multidimensi merupakan tabel yang berisi informasi yang diatur dalam baris dan kolom.
- Untuk mengidentifikasi elemen array tersebut maka kita harus menambahkan :
 - Indeks yg menyatakan elemen baris dan
 - Indeks kedua yg menyatakan elemen kolom

Array yg menggunakan dua indeks untuk mengidentifikasi element tertentu disebut array dua dimensi,

Visual basic mendukung sedikitnya 60 array

- Visual basic mendukung sedikitnya 60 array namun kebanyakan penggunaanya memerlukan maksimal dua atau tiga dimensi

Contoh definisi array multidimens

Bentuk

Dim <VariableName> (n1, n2) as <DataType>

n1 = Baris n2 = Kolom

Dim nama (5,5) as String

Dim alamat(1 To 5, 1 To 5) As String

Contoh array multidimens

```
Dim kata(1 To 2, 1 To 2) As String
kata(1, 1) = "Halo"
kata(1, 2) = "Apa Kabar"
kata(2, 1) = "Selamat Pagi"
Contoh Contoh array array multidimensi multidimensi
kata(2, 1) = "Selamat Pagi"
kata(2, 2) = "Hai.."
Debug.Print kata(2, 2) & " " & kata(1, 2)
```

Hasilnya

Hai apa kabar

Array dan Variant

Dalam visual basic memungkinkan kita menyimpan array ke dalam variabel bertipe variant, dan kemudian mengakses variabel variant tersebut sebagai array.

```
Dim nama(3) As String
```

```
Dim tampung As Variant
```

```
tampung = nama
```


Contoh

Dim nama(3) As String

Dim tampung As Variant

nama(0) = "santoso"

nama(1) = "Abdilah"

nama(2) = "Nurul"

nama(3) = "Udin"

tampung = nama

Debug.Print tampung(2)

Hasilnya : Nurul

Kita juga dapat membuat array dalam elemen variant menggunakan fungsi Array dan menyimpannya ke dalam variabel variant

```
Dim Factorials as Variant
```

```
Factorials = Array(1, 1, 2, 6, 24,  
120, 720, 5040, 40320, 362880,  
3628800)
```

```
Debug.Print (Factorials(1))
```

Prosedur

- Prosedur dapat kita definisikan sebagai suatu program yg terpisah dalam blok sendiri yg berfungsi sebagai sub program (program bagian)
- Visual basic menawarkan berbagai jenis prosedur untuk melaksanakan sebagian kecil coding dalam aplikasi.

Pengertian Prosedur

- Prosedur berguna untuk kondensasi operasi yang berulang seperti perhitungan yg sering digunakan, operasi yg sering diulang, dsb.
- Beberapa manfaat menggunakan prosedur sbb:
 - Lebih mudah untuk men-*debug* program yg dipecah kedalam bagian yg lebih kecil
 - Prosedur yg digunakan dlm satu program dapat bertindak sebagai *building block* untuk program lain dengan sedikit modifikasi.

Tipe Prosedur dalam Visual Basic

Prosedur dalam visual basic 6 dapat dibagi menjadi beberapa prosedur sbb:

1. Sub Procedure (Prosedur Sub)
2. Event Procedure (Prosedur Event)
3. Function Procedure (Prosedur Fungsi)

1. Sub Procedure

- Merupakan prosedur standar yg berfungsi sebagai *sub rutin* yg mengerjakan perintah-perintah tertentu setiap terjadi pemanggilan
- diawali dengan keyword **Sub** dan diakhiri dengan **End Sub**.
- Prosedur ini dapat ditempatkan dalam *form, class,* atau *module*.

Sintak Sub Procedure

[Private | Public] [Static] Sub <nama_sub> [(argumen)]

[statements]

End Sub

1. Sub Procedure

- Pernyataan [**Public**| **Private**] menentukan ruang lingkup (scope) *function*. Sebuah *function* dengan scope public digunakan dalam lingkup project. Sedangkan *function* dengan scope private hanya bisa digunakan dalam lingkup form saja.
- <**nama_sub**> dibuat sebagai pengenalan procedure saat di-'panggil'. Aturan penamaan sebuah procedure sama dengan aturan penamaan sebuah variabel. Nama sebuah procedure dibuat unik, tidak boleh ada yang sama.
- <**argumen**> merupakan serangkaian nilai dan tipe data yang dipakai oleh procedure untuk mengerjakan 'tugas'-nya atau sering disebut sebagai paramater. Sebuah procedure bisa saja tidak memakai argumen sama sekali.

Contoh Sub Procedure

Private Sub aktif()

```
txtnim.Enabled = True  
txtnama.Enabled = True  
txtalamat.Enabled = True  
cmbkota.Enabled = True
```

End Sub

Private Sub Form_Load()

aktif

Call aktif

End Sub

Pemanggilan Prosedur, boleh langsung **dipanggil** nama prosedur atau menggunakan **keyword Call**

2. Event Procedure

- **Event procedure** merupakan prosedur yg diaktifkan oleh system sebagai response terhadap *event*.
- Sebuah **Event Procedure** merupakan sebuah blok yg berisi nama sebenarnya suatu kontrol atau komponen, dan nama *event* yg akan diresponse setelah sebelumnya dihubungkan dengan *underscore* (_)

Contoh Event Procedure

```
Private Sub Form_Load()
```

```
-
```

```
-
```

```
End Sub
```

```
Private Sub Command1_Click()
```

```
-
```

```
-
```

```
End Sub
```

Default *scope* pada *event procedure* adalah ***Private***

3. Function Procedure

- *Function procedure* seperti *sub procedure* yang membedakan adalah pada *function procedure* memiliki mekanisme pengembalian nilai ke prosedur pemanggilnya sedangkan *sub procedure* tidak.
- *Function procedure* memiliki tipe data seperti halnya variabel
- *Return value* (nilai pengembalian) dimasukkan ke dalam fungsi itu sendiri sehingga suatu fungsi dapat menjadi bagian dari ekspresi yg panjang.

Sintak Function Procedure

```
[Private | Public] [Static] Function Functionname [(argumen)] as  
tipe _data>  
    [statements]  
End Function
```

Keterangan :

- Pernyataan [Public | Private] menentukan ruang lingkup (scope) function. Sebuah function dengan scope public bisa digunakan dalam lingkup project.

Sedangkan function dengan scope private hanya bisa digunakan dalam lingkup form saja.

Sintak Function Procedure

- <nama_function> dibuat sebagai pengenalan function saat di-'panggil'. Aturan penamaan sebuah function sama dengan aturan penamaan sebuah variabel. Nama sebuah function dibuat unik, tidak boleh ada yang sama.
- <argumen> merupakan serangkaian nilai dan tipe data yang dipakai oleh function untuk mengerjakan 'tugas'-nya atau sering disebut sebagai paramater. Sebuah function bisa saja tidak memakai argumen sama sekali.
- Pernyataan As <tipe_data> pada function menentukan tipe data nilai yang akan dikembalikan (return value) setelah 'tugas'-nya selesai.

Contoh Function Procedure

Function tambah(bil1 As Single, bil2 As Single)

tambah = bil1 + bil2

End Function

Private Sub Form_Load()

Debug.Print tambah(6, 4)

End Sub

Function Pangkat(Bilangan as Integer)

Pangkat = Bilangan * Bilangan

End Function